Topics in Machine Learning (TIML-10)

Instructor: Saketh

Contents

Contents **i**

1 Lecture 1 **3**

1.1 Summary 3

1.2 Further Reading 3

2 Lecture-2 **5**

2.1 Summary 5

2.2 Further Reading 6

3 Lecture-3 **7**

3.1 Summary 7

3.2 Further Reading 8

4 Lecture-4 **9**

4.1 Summary 9

4.2 Further Reading 9

5 Lecture-5 **11**

5.1 Summary 11

5.2 Further Reading 12

6 Lecture-6 **13**

6.1 Summary 13

6.2 Further Reading 14

7 Lecture-7 15

7.1 Summary 15

7.2 Further Reading 16

8 Lecture-8 17

8.1 Summary 17

8.2 Further Reading 18

9 Lecture-9 19

9.1 Summary 19

9.2 Further Reading 20

10 Lecture-10 21

10.1 Summary 21

10.2 Further Reading 21

11 Lecture-11 23

11.1 Summary 23

11.2 Further Reading 24

12 Lecture-12 25

12.1 Summary 25

12.2 Further Reading 26

13 Lecture-13 27

13.1 Summary 27

13.2 Further Reading 28

14 Lecture-14 29

14.1 Summary 29

14.2 Further Reading 29

15 Lecture-15 **31**

15.1 Summary 31

16 Lecture-16 **33**

16.1 Summary 33

16.2 Further Reading 34

17 Lecture-17 **35**

17.1 Summary 35

18 Lecture-18 **37**

18.1 Summary 37

19 Lecture-19 **39**

19.1 Summary 39

19.2 Further Reading 40

20 Lecture-20 **41**

20.1 Summary 41

21 Lecture-21 **43**

21.1 Summary 43

21.2 Further Reading 44

22 Lecture-22 **45**

22.1 Summary 45

22.2 Further Reading 46

Lecture 1

1.1 Summary

This lecture is an introduction to the Statistical Learning Theory (STL). Using the knowledge of probability theory we pose the problem of picking a "good" learning function as that of minimizing the *Risk* functional. The Risk functional is impossible to evaluate whereas an estimate of *Risk* can be easily obtained using the training data. Hence we suggest "minimizing the *Empirical Risk* functional" instead (for some loss functions this minimization itself may be NP-hard). Using notions of convergence of empirical processes, we try to arrive at conditions where "Empirical Risk Minimization (ERM)" is "good"; in sense that it is "as good as" minimizing the true Risk functional. We concluded the lecture by mentioning the key theorem in STL by Vapnik and Chervonenkis [1991]. Discussion of this theorem will start the next lecture.

1.2 Further Reading

Here are some pointers to further readings:

Math Basic math used in this lecture requires knowledge in these topics: a) notions of expectation and convergence of random variables b) law of large numbers c) Concentration Inequalities. Good reference for topics a) and b) is Saketh [2009] (lectures 8, 9,22, 23, 24) and for c) is Boucheron et al. [2004].

ML We covered pages 125–134 in Schölkopf and Smola [2002].

Lecture 2

2.1 Summary

After revising the concepts explained in the previous class, we attempted to formalize the notion of consistency of the principle of ERM. Basically we saw that in cases where there exists a global minimizer for the loss function, i.e. $\exists f^* \in \mathcal{F} \ni l(f^*(X), Y) \leq l(f(X), Y) \; \forall \; f \in \mathcal{F}$, consistency of ERM is trivially achieved. Though we did not go into further details, we noted that there are notions of "non-trivial" consistency, the necessary and sufficient conditions of which are sought for.

For the purposes of this course we say ERM is consistent iff in probability

$$\inf_{f \in \mathcal{F}} R^m_{emp}[f] \to \inf_{f \in \mathcal{F}} R[f]$$

Subsequently, we proved that one-sided uniform convergence:

$$P[\sup_{f \in \mathcal{F}} R[f] - R^m_{emp}[f] > \epsilon] \to 0, \; \forall \; \epsilon > 0$$

as $m \to \infty$ is a sufficient condition for consistency (refer supplementary for the proof) and noted that for "non-trivial" consistency these are infact necessary. This completed the discussion of the key theorem in learning theory due to Vapnik and Chervonenkis [1991] mentioned in last class.

Once this is done, we saw that in cases where $\mathcal{F}$ has finite number of functions, the ERM is consistent. This involved simple application of union and chernoff bounds. However this wont work if number of functions is infinite. The following is a key observation which gives us a way to handle generic cases: however large $\mathcal{F}$ is, on a finite set of examples (say training examples), many of the functions look the same. Infact, for the case of binary classification, on m points there cannot be more than 2^m different functions. Thus though the functions are different, on a finite sample they essentially look the same! The idea is to exploit this observation and the only term hindering the development is the true risk term in the uniform convergence criterion. This term is eliminated using the trick of symmetrization or ghost sampling which gives: $P[\sup_{f \in \mathcal{F}} R[f] - R^m_{emp}[f] > \epsilon] \leq 2P[\sup_{f \in \mathcal{F}} \hat{R}^m_{emp}[f] - R^m_{emp}[f] > \epsilon/2]$. Now since the RHS involves only empirical terms i.e. deals with $2m$ (finite) examples, the class $\mathcal{F}$ would essentially look finite — the number of functions is given by notions of shattering coefficient etc. defined in the lecture. Using shattering coefficient, union and (modified) Chernoff bound, one can arrive at the conclusion that as long as the shattering coefficient does not exponentially grow with m, consistency of ERM is guaranteed. The lecture ended with a hint that hyperplane classifiers do satisfy this in some sense. More discussions on this would constitute the next lecture.

2.2 Further Reading

- Read Schölkopf and Smola [2002] pages 131–136.
- For definition of non-trivial consistency (also empirical processes) refer Vapnik [1998] pages79-86.
- Simple proof for symmetrization and final bound using union and (modified) chernoff tricks is in Bousquet et al. [2004], section 4.4.
- Shattering coefficient etc. is explained well in Burges [1998].

Lecture 3

3.1 Summary

In detail, with examples, we studied notions of shattering coefficient $\mathcal{N}(\mathcal{F}, m)$, the variant of $\mathcal{N}$ which depends on the training sample $\mathcal{N}(\mathcal{F}, Z_m)$, growth function $\mathcal{G}(\mathcal{F}, m)$. Using the notion of shattering coefficient, it was then easy to generalize the bound obtained in case of finite $\mathcal{F}$ to the case of $\mathcal{F}$ having infinite functions. This analysis gave the sufficient conditions for consistency of ERM (in the case of indicator loss functions) to be: $\lim_{m\to\infty} \mathcal{G}(\mathcal{F}, m)/m = 0$. Infact, we noted that this condition is necessary and sufficient for exponentially fast rate of convergence in context of ERM (ofcourse in the case of indicator loss functions) irrespective of what the underlying probability distribution is! Encouraged by this, we went ahead and looked at a related concept of VC dimension which would neatly summarize the concept of "capacity" of a set of functions. We saw an important relation between growth function and VC dimension [Vapnik and Chervonenkis, 1971] which helps us make the following strong statement: "For the special case of binary classification problems with 0-1 loss functions, VC dim. being finite implies exponentially fast convergence in context of ERM and VC dim. being infinite implies no exponentially fast convergence".

Subsequently we re-wrote the bounds obtained as what are known as VC-type inequalities: "with probability $1-\delta$, we have $R[f] \leq R_{emp}[f] + \Phi(m, h, \delta)$" (the Φ term is called the confidence term). A closer look at VC-inequalities motivated the *Structural Risk Minimization* (SRM) principle; where instead of minimization empirical risk alone, one minimizes the sum of the empirical risk and confidence terms. After outlining SRM, we took the specific case of canonical hyperplane classifiers and saw how implementing the SRM leads to the famous maximum-margin principle of Support Vector Machines (SVMs).

The bounds derived previously work with any underlying distribution, which is both the greatest advantage and disadvantage of them. Data-dependent bounds, which work only with the specific underlying (unknown) distribution, may be are more tighter. Employing the quantity $\mathcal{N}(\mathcal{F}, Z_m)$ inplace of shattering coefficient is obvious for this purpose; however the difficulty is that $\mathcal{N}(\mathcal{F}, Z_m)$ is a random quantity in itself. This problem was eleviated by using the notion of conditional expectation and introducing Rademacher variables as dummy variables to condition on. The bounds thus derived lead to the definition of annealed entropy $\mathcal{N}_{Ann}(\mathcal{F}, m)$. The necessary and sufficient conditions for exponentially fast convergence with the specific underlying probability distribution turns out to be $\lim_{m\to\infty} \mathcal{N}_{Ann}(\mathcal{F}, m)/m = 0$. Since annealed entropy is difficult to compute (as underlying distribution is unknown), the idea of Rademacher averages is introduced. The lecture ended with a very very brief introduction to Rademacher averages for function classes.

3.2 Further Reading

- Pages 137-142 in Schölkopf and Smola [2002]
- In Bousquet et al. [2004], read section 3 for topics covered in this lecture. Interested students can read sections 5.2, 6.4 in the same.
- For derivation of important relation between growth function and VC dim, look into section 4.10 in Vapnik [1998]. Another version of this proof is in Ben-David [2003]. An easy proof is here Smolensky [1997].
- Some further reading on VC dimension: Burges [1998], Sontag [1998] and works on VC dim. stuff by Peter Bartlett[1].
- Conditional expectation is explained in Lectures 18,19 [Saketh, 2009].

[1] At `http://www.stat.berkeley.edu/~bartlett/publications/pubs-93.html`

Lecture 4

4.1 Summary

One of the main take-home from previous lecture was the fact that, in case of binary classification problems (with 0-1 loss functions), ERM is consistent with exp. fast convergence if and only if VC dimension is finite. Moreover, according to SRM, it is desirable to have low VC dimension while doing well on the training dataset. Given this and the fact that hyperplane classifier's VC dim. grows linearly with the dimensionality of the data, it may not recommendable to employ even "simple" classifiers like the hyperplane classifiers for data in very high dimensions. With this observation, we seek a modified class of hyperplane classifiers whose VC dimension (ideally) does not grow with the dimensionality of the problem.

We study the class of canonical hyperplane classifiers whose VC dimension can be shown to be independent of the dimensionality! These classifiers enforce a separation of positive and negative data points with some non-zero margin. We proved an important theorem [Bartlett and Shawe-Taylor, 1999] which showed that the VC dim. of canonical hyperplane classifiers is $\propto W^2R^2$ (and hence independent of the dimensionality). Using this result and the principle of SRM, we noted the famous (hard-margin) *Support Vector Machine* (SVM) formulation, which is an instance of a (regular) convex Quadratic Program (QP) and hence efficiently solvable. We ended the lecture with some discussion of convex functions and convex optimization problems.

4.2 Further Reading

- Pages 142-146 in Schölkopf and Smola [2002]
- More on the theorem: Shawe-Taylor et al. [1998]

- Jensen's inequality: Lecture 9 [Saketh, 2009]
- On SOCPs: Lobo et al.

Lecture 5

5.1 Summary

The lecture started with a brief review of the theorem on VC dim. of canonical hyperplane classifiers and its use in motivating the SVM formulation through the application of SRM. After a little thought it was clear that we (purposefully) overlooked a technicality and the theorem cannot be directly applied under the framework of SRM. This is because, the VC dim. as we derived in last lecture is dependent on the training set of examples given! We noted that in case of all the CV-type inequalities derived till now, the confidence term is either data-independent or data-dependent; in the sense that they are either valid for all prob. distributions or the particular (unknown) prob. distribution under consideration. However in this case since VC dim. is dependent also on the particular sample (training set) from the prob. distribution and hence the bounds do not apply directly. This is what motivates the principle of *data-dependent SRM* [Shawe-Taylor et al., 1998, Shawe-Taylor and Cristianini, 1998] where the structure of the learning functions (i.e. arrangement of the learning functions in $\mathcal{F}$ in non-decreasing order of complexity) itself could be dependent on the training set[1].

Subsequently we looked at a result by Bartlett and Shawe-Taylor [1999] which provides a VC-type inequality suitable for data-dependent SRM. Using this result we derived both the hard-margin and soft-margin versions of the SVM formulations. In the course of the derivation we also discussed notions of Ivanov [Ivanov, 1976], Tikhonov [Tikhonov and Arsenin, 1977] and Morozov [Morozov, 1984] regularizations and the concept of hinge-loss function.

We concluded the lecture with a brief introduction to Lagrange multiplier and duality theory.

[1] Note the two different senses in which *data-dependent* term is used: once for saying specific to underlying prob. distribution and once for being specific to the training sample

5.2 Further Reading

- Pages 189–195 in Schölkopf and Smola [2002] and other references cited in the summary
- Duality theory: Section 6.3 in Schölkopf and Smola [2002], sections 5.1–5.5 in Boyd and Vandenberghe [2004]
- Soft-margin SVMs were proposed in Cortes and Vapnik [1995]. This is actually the second paper in SVMs.

Lecture 6

6.1 Summary

We briefly revised the duality theory and (Karush-Kuhn-Tucker) KKT conditions for optimality. Subsequently we derived the dual of the hard-margin SVM formulation and wrote down the KKT conditions (which are in this case necessary and sufficient). This analysis gave some key insights: i) at optimality, $\mathbf{w}$ is a linear combination of the training points ii) moreover, the linear combination is sparse in sense that only few of the training points are involved iii) geometrical interpretation of dual is that of minimizing distance between convex hulls of positive and negative datapoints (Derivation for soft-margin SVM case is here Bennett and Bredensteiner [2000]) iv) both while training the SVM and while making predictions knowledge of dot-products involving the datapoints are enough; the actual feature representation of them is not needed. Interestingly, these insights were exploited in order to arrive at efficient solvers (which are faster than generic interior point methods) for the SVM problem: a) SMO algorithm [Platt, 1999, Keerthi et al., 2001] which exploits the fact that the solution for dual is sparse b) Nearest point alg. for min. distance between convex hulls [Keerthi et al., 2000].

Till now we learned that SVMs are supported by rigorous statistical theory as well as enjoy computationally efficient solving techniques. However, the generalization ability would still be restricted because after all the set of classifiers realizable are linear discriminators. Through assignment problems we note that *polynomial discriminators* are “richer” class of functions than linear discriminators and one can give numerous examples of data where linear discriminators are inherently very restrictive. With this observation, we set out with a dream of coming up with a framework where we can benefit from advantages of canonical hyperplane classifiers (i.e. VC dim. being dimension independent) while employing polynomial or in general, non-linear discriminators.

As a first step, we saw that d degree polynomial discriminators are essentially linear discriminators in a higher dimensional space expanded using all possible monomials constructed with input features of degree d. Further, we saw that the dot-products in the exanpded feature space can be computed using usual dot-product in input space. Hence computationally this trick of dealing with polynomial discriminators as linear ones is feasible as SVMS require knowledge of dot-products alone! Encouraged by this, we seek to extend these ideas for dealing with generic input spaces (which need not be vector-spaces) and generic non-linear discriminators using the notion of *kernels*.

6.2 Further Reading

- Derivation of dual and geometrical interpretation: section 2 in Keerthi et al. [2000]
- Pages 25–30 in Schölkopf and Smola [2002] and other references cited in the summary
- Chp. 2 in Shawe-Taylor and Cristianini [2004]

Lecture 7

7.1 Summary

We reviewed the definition of positive kernel and some of the implications of it. We then noted that if input-space is Euclidean, then the dot-product is indeed a kernel (we call this linear kernel). Then we went on to examples which made it clear that as long as the given function is an inner-product in some Euclidean space, then it is a kernel e.g. *polynomial kernels.* We then wrote down a rough version of Mercer's theorem [Mercer, 1909] which basically assured that given an inner-product it is a kernel and given a kernel there exists some (Hilbert) inner-product space where kernel is an inner-product. We noted a rough constructive proof of the latter part of the claim. The construction gave many insights: i) each element of input-space is mapped to a function which measures similarity between the element and all others using the kernel itself ii) This functional representation provides a very rich description of elements of input-space iii) a ("smallest") inner-product space containing the "feature representations" of these elements is then constructed such that the inner product in that space is the given kernel.

Subsequently we noted the equation of a hyperplane in the feature space and with examples of some kernels argued that a hyperplane discriminator in feature space would essentially be a non-linear discriminator in the input space. The concept of Gaussian kernel was introduced as a natural extension of polynomial kernels. We also noted that normalization, sum, products of kernels is again a kernel. A theorem by Micchelli [1986] helped us to argue that a Gaussian kernel essentially maps data to an infinite dimensional space and represents a non-linear discriminator in the input space.

7.2 Further Reading

- Pages 31-36 in Schölkopf and Smola [2002]
- Theorem on Gaussian kernels: thm.2.8 in Schölkopf and Smola [2002]
- First use of kernel trick: Aizerman et al. [1964] and second use (first paper on SVMs): Boser et al. [1992]

Lecture 8

8.1 Summary

Lecture started with a review of the key theorem discussed in previous lecture regarding positive kernels. We then briefed the actual Mercer theorem [Mercer, 1909] and summarized the main take-home: a) Mercer's theorem presented another inner-product space (actually an RKHS) where the given kernel is an inner-product b) There is no unique RKHS for a given kernel (though uniqueness for converse statement can be proved) c) This representation had obvious links like Euclidean spaces (and hence notions of ortho-normal basis etc. are immediate); however the mapping ϕ itself is not intuitive. On the contrary, for the representation discussed in the previous lecture, the mapping ϕ was very intuitive; whereas notions of ortho-normal basis etc. were not obvious (atleast for us). Subsequently we proved that for any arbitrary inner-product space, the inner-product is a positive kernel. To summarize, the key points to note are: i) given a positive kernel there exits are RKHS in which the kernel is an inner-product (and hence measures similarity) and vice-versa ii) the representation of input-space objects in this RKHS is very rich!

Later on, we completed our discussion on operations preserving positivity of kernels: in particular, we showed that sum, scaling, product, polynomials, exponential of positive kernels is again a positive kernel. We gave examples of kernels not defined on Euclidean spaces esp. dealing with probability spaces: i) kernel on event space of a random expt. with ϕ as mapping $\phi(A) = 1_A - P(A)$ and the inner-product space as the usual vec. spa. of mean zero rvs., endowed with $E[XY]$ as the inner-product. ii) probability product kernels [Jebara et al., 2004]. We saw that when all the training points are assumed to be noisy with spherical covariance and Normally distributed then the prob.prod. kernel is nothing but the Gaussian kernel. This throws more light on the usefulness/appropriateness of Gaussian kernel in Kernel methods.

Subsequently we derived the dual of the SVM formulation built in an RKHS corresponding to a given kernel. This was done using what is known as the Representer theorem [Schölkopf et al., 2001] (we will state the general version in next lecture). We noted that by virtue of theorem discussed in last class, the gram matrix of training examples can never be rank-deficient (i.e., gram-matrix is always positive definite) when the Gaussian kernel is employed (assuming there are no duplicates in the training examples) and hence the solution of soft-margin SVM is unique in this case. We also noted that the primal of hard-margin SVM always has a unique solution.

8.2 Further Reading

- Pages 36–39 in for Mercer kernels and 89–91 for representer theorem in Schölkopf and Smola [2002] and citations in above summary
- Supplementary for derivation of dual
- For uniqueness arguments for SVM refer Burges and Crisp [2000]
- For various other eg. of kernels NOT defined on Euclidean spaces see chp. 9-12 (part-3) of Shawe-Taylor and Cristianini [2004]

Lecture 9

9.1 Summary

The issue of recovering the hyperplane parameters $(\mathbf{w}, b)$, and in particular, determining $f(\mathbf{x}) = \text{sign}(\mathbf{w}^\top \mathbf{x} - b)$ for a test datapoint $\mathbf{x}$ from the solution of the dual of SVM (i.e, from optimal Lagrange multipliers) was discussed. Notions of bounded and non-bound support vectors etc. were clarified. It was noted that in case the gram-matrix of training datapoints was non-singular, the term $\mathbf{w}^\top \mathbf{x}$ is determined uniquely and b can be computed using the margin of support vectors (refer supplementary). In the case where the gram-matrix is singular, the recovery is not unique and can be chosen such that computation of $f(\mathbf{x})$ involves as few support vectors as possible. Lesser no. support vectors is desirable as it implies better efficiency at prediction stage.

The SLT results for linear discriminators (with non-zero margin) can be extended to many loss functions (other than the 0-1 loss function we always considered). We donot venture into the details, but nevertheless it is easy to imagine this will again lead to solving optimization problems which will minimize the empirical loss and the complexity term $||\mathbf{w}||$. We noted that the representer theorem can easily be extended to handle these cases. Hence all of these methods can be kernelized: e.g., ridge-regression [Hoerl, 1962] (loss function is $\left(y - (\mathbf{w}^\top \mathbf{x} - b)\right)^2$), SVM-regression [Drucker et al., 1996, Smola and Schölkopf, 2004] (loss function is ϵ-insensitive loss: $\max(0, |y - (\mathbf{w}^\top \mathbf{x} - b)| - \epsilon)$ or Huber loss) etc. (see also table 3.1 in Schölkopf and Smola [2002]). Further, some loss functions like the hinge loss, ϵ-insensitive loss etc., promote sparse solutions, i.e., the prediction $f(\mathbf{x})$ can be computed using relatively small number of training datapoints usually called as Support Vectors — and hence are known as "Support Vector Methods".

Also, motivated by the equivalence of norms in Euclidean spaces, once can try replacing the usual $||\mathbf{w}||_2$ (regularization/complexity term) by other norm-based regularizers like $||\mathbf{w}||_1$, $||\mathbf{w}||_\infty$ etc. It was noted that in general such formulations cannot be kernelized however may be employed in practice owing to the merits of these regularizers: for example, $||\mathbf{w}||_1$ promotes sparsity in optimal $\mathbf{w}$ and hence is suitable for feature selection tasks, similarly $||\mathbf{w}||_\infty$ promotes equal values for components of $\mathbf{w}$ at optimality and hence is suitable for situations where all features are known to be more-or-less equally important.

The lecture concluded with a brief discussion of choosing model parameters like C (regularization parameter) and other kernel parameters. Methods based on validation sets, minimizing upper bounds on risk were briefed.

9.2 Further Reading

- E.g. of works which try to minimize/control the number of support vectors (with intention of reducing computational effort in prediction stage): Burges and Schölkopf [1997], Schölkopf et al. [2000], Keerthi et al. [2006] and references therein.

- Look at chp.3 in Schölkopf and Smola [2002] for discussion on alternative loss functions and the subsequent chapter for alternative regularizers.

- Chapelle et al. [2002] is an example of work where the idea is to choose model parameters by minimizing various bounds on risk. At each step an SVM problem is solved.

Lecture 10

10.1 Summary

Ever since the notion of kernels was introduced, we felt the need for exploring ways of automatically coming up with the kernel which suits a given application (training dataset). The need is furthered by observations from various applications that the performance of SVMs critically depends on the choice of kernel employed. Hence we set out to address the problem of Kernel Learning — simultaneously optimize for the right kernel as well as the discriminating function (classifier) given the training data.

In order to motivate an interesting way of formulating the Kernel Learning problem, we revisit the SLT and derive new learning bounds which are based on an alternate way of computing complexity of a set of learning functions (rather than VC dim., covering no. etc.) — known as Rademacher Complexity or Rademacher Average [Kolchinskii et al., 2001, Bartlett and Mendelson, 2002]. As a first step we derived the McDiarmid's inequality [McDiarmid, 1989].

10.2 Further Reading

- Bousquet et al. [2004], Mendelson [2003] are good references for the derivation in this lecture.

Lecture 11

11.1 Summary

Notion of Rademacher average (Rademacher complexity, $\mathcal{R}$) was introduced and learning bounds involving $\mathcal{R}$ were derived. The major steps involved are: i) noting that $T = \max_{f\in\mathcal{F}}\left\{R[f] - R^m_{emp}[f]\right\}$ is a function of independent (infact, iid) random variables $Z_1,\ldots,Z_m$ satisfying the bounding difference property (consequence of assuming loss function is bounded) and then upperbounding bounding $P(T-\mathbb{E}[T] > \epsilon)$ using McDiarmid's inequality. This leads to a VC-type inequality where true risk is upper bounded by sum of empirical risk, $\mathbb{E}[T]$ and confidence terms, ii) upper bounding $\mathbb{E}[T]$ by twice $\mathcal{R}$ of induced set of loss functions. iii) upper bounding $\mathcal{R}$ by sum of conditional R.A. ($\mathcal{R}_m$ — which is computable) and confidence term.

In case of binary classification (with 0-1 loss), the bound was further written in terms of $\mathcal{R}_m$ of $\mathcal{F}$ itself. Intuition for $\mathcal{R}_m$ being a measure of richness of set of classifiers was given and bounds relating $\mathcal{R}_m$ to VC-dim. were presented. It was noted that the new bounds are "tighter" than the usual VC-dim. based bounds derived earlier.

Encouraged by this, we presented a neat upper bound on $\mathcal{R}_m$ of the set of affine functions in RKHS (associated with a given kernel k) which are norm bounded and are linear combinations of images of training datapoints in feature space (recall representer theorem): $\mathcal{R}_m(\mathcal{F}) \leq \frac{W}{m}\sqrt{trace(K)}$, where K is the gram-matrix of training datapoints with kernel k. This will motivate a good kernel learning formulation, which will be studied in the next lecture.

11.2 Further Reading

- Refer Bartlett and Mendelson [2002] for detailed derivation of the trace bound.

Lecture 12

12.1 Summary

A small technical note on bounding $\mathcal{R}_m$ of real-valued functions rather than $\{-1, 1\}$-valued functions was presented. Based on the margin-trace bound derived in the previous lecture, a kernel learning formulation was motivated [Lanckriet et al., 2002, 2004]. The problem can be posed as an SDP (Semi-Definite Program) and solved. However solving this formulation gives the optimal gram-matrix for training datapoints and the corresponding optimal hyperplane. Prediction is impossible because the kernel evaluated at any test and training datapoint is unknown. Realizing the importance for coupling the training adn test datapoints, it was proposed to search the space of all trace bounded conic combinations of a given set of base (positive) kernels rather than all kernels which are trace bounded and psd. It is easy to see that predictions using this methodology can be done, once the problem is re-written in terms of these kernel weights, and solved for the unknowns. Also, from practical perspectives the methodology is useful: i) tuning kernel parameters ii) combining benefits of carefully designed kernels iii) Multi-modal tasks. Researchers from various fields like vision and bio-informatics have observed encouraging results by employing this new formulation. Since the formulation learns the combination of kernels as wells as the corresponding classifier, it is called as Multiple Kernel Learning (term coined by Bach et al. [2004]). The problem of MKL was posed as a quadratically constrained linear program, which can be solved efficiently using off-the-shelf solvers. This derivation gave more insights into the nature of the solution: at optimality essentially the single "best" kernel of the given base kernels is picked!

12.2 Further Reading

- Refer Bartlett and Mendelson [2002] for detailed discussion of technical note presented at start of lecture.
- Vandenberghe and Boyd [1996] is a good reference for SDPs.
- Sion [1958] is paper on minimax theorem used in derivation.

Lecture 13

13.1 Summary

In this lecture we studied the various ways in which researchers have attempted to solve the MKL formulation. Since the MKL essentially selects few kernels from the given base kernels, it can be employed for non-linear feature selection (construct base kernels using individual features!). Hence algorithms for solving the MKL need to be scalable wrt. no. base kernels as well as no. training examples in order to perform efficient feature selection. The methods suggested in the original paper [Lanckriet et al., 2004] and in Bach et al. [2004] are not scalable to large datasets. Hence researchers attempted to solve this formulation using efficient first-order techniques.

Sonnenburg et al. [2006] pose the MKL problem as a Semi-Infinite Linear Program (SILP) and solve it efficiently using a cutting-plane algorithm (exchange algorithm in context of SILP). Essentially, the algorithm solves an regular SVM problem and a LP (Linear Program) in each iteration. Since this algorithm re-uses SVM code, it can be efficient if efficient SVM solvers are employed. Cutting-plane algorithms achieve global convergence; however convergences rates are unknown (in general). The proposed algorithm infact takes large no. iterations (large no. SVMs solved) to converge in this case. Taking this method as inspiration, subsequently many have attempted to solve MKL by solving series of SVMs; however with as less number of SVM calls as possible. Sonnenburg et al. [2006] also propose few modifications to regular SMO-based SVM solvers which further increases the efficiency of the methodology. Refer the paper for further details.

Rakotomamonjy et al. [2007, 2008] propose to solve MKL by posing it as a problem of minimizing a convex (Lipschitz conts.) function (whose gradient can be computed using Danskin's theorem) over a simplex by employing reduced-gradient technique. We briefly reviewed gradient-descent methods for unconstrained and constrained optimization problems. Also we saw how Danskin's theorem can be applied in case of MKL. Under proper choices of step-sizes, gradient-descent algorithms are gauranteed to converge and infact rates of convergence are also established.

13.2 Further Reading

- Refer sec. 1.1-1.3, 2.1-2.3, 6.3.3 in Bertsekas [1999] for some efficient first-order methods
- Also in Luenberger and Ye [2008] chp.7,8 for revising basics and sec.12.4-12.9 for gradient-based methods including reduced-gradient method; sec.14.7 for cutting-plane alg.
- Danskin's theorem — prop. B.25 in Bertsekas [1999]

Lecture 14

14.1 Summary

We saw how to apply the projected gradient descent (*pgd*) and reduced gradient descent (*rgd*) algorithms to the MKL formulation. Both methods have proofs of convergence and infact, bounds on rates on convergence can be derived. At every iteration, *pgd* solves an SVM problem for calculating gradient and involves a projection onto a simplex. The step-size can be chosen easily (diminishing step-sizes). Hence per-iteration cost is atleast $O(SVM) + O(d^2)$. *rgd* on the other hand solves an SVM for obtaining gradient and solves few more SVM problems for evaluating the objective during step-size selection with Armijo's rule. So per-iteration cost is solving few SVM problems. In practice the training time of both the methods is comparable and far lower than that with cutting plane methods derived in previous lecture. Also these methods are more scalable wrt. n.

We then introduced a variant of *pgd* which is known as *Mirror-descent* [Ben-Tal et al., 2001, Beck and Teboulle, 2003]. The idea is to employ a Bregmann Divergence based regularization term rather than a Euclidean-norm based term (in order to constrain that the next iterate is close to the prev. iterate) such that the projection problem is "easy" to solve. Infact, for the case of simplex, if the Bregmann divergence is selected as KL-divergence, then this problem has a closed form solution which can be easily computed; these details we will see in the next lecture.

14.2 Further Reading

- Refer sec. 5.3.1 in Nemirovski [2005] for review of *pgd* and sec. 5.4.1, 5.4.2 for nice explanation of mirror descent. Also look into sec. 5.5.1.

Lecture 15

15.1 Summary

Application of Mirror-Descent (MD) to the problem of MKL was the main focus of the lecture. Since the feasibility set is a simplex (a standard set-up well-known in optimization literature), the entropy function was chosen as the generating function. With such a choice we showed that the per-step optimization problem (which involves projection) is easy to solve i.e., the values of λ can be updated in $O(n)$ time (compared to $O(n^2)$ for *pgd* which employs Euclidean regularizer). The algorithm is also expected to be faster than *rgd* as it employs "diminishing" step-sizes instead of Armijo's rule (which is very costly in our case as it involves few SVM calls for objective function evaluation). We also noted that MD has good convergence rates: if step-sizes are chosen in a particular "optimal" way (exact formulae have been noted in lecture or refer Nemirovski [2005]), then the number of iterations required for the deviation in objective from true optimal objective to be less than ϵ is proportional to $\log(n)/\epsilon^2$, which is good news as one can approximately solve problems with high-dimenisonality easily. MD can be applied to any optimization problem of the form $\min_{\mathrm{x}\in\mathcal{X}} f(\mathrm{x})$ where:

1. f is convex
2. $\mathcal{X}$ is compact
3. Oracle which gives (sub-)gradient of f exists (f itself may not be computable/known) — this is known as "black-box" setting
4. f is Lipschitz

In such settings with $\mathcal{X}$ being a simplex, one can show that in some sense, one cannot beat the MD algorithm! Standard set-up's for MD are: $\mathcal{X}$ being a sphere, simplex, full-simplex, spectrahedron, full-spectrahedron. MD when applied to

MKL performs far better than `simpleMKL` — we saw some simulation results supporting this [Nath et al., 2009]. For sake of completeness of discussion of various strategies of solving MKL we noted methods of Chapelle and Rakotomamonjy [2008], Xu et al. [2008].

We recalled two reasons why MKL basily is a kernel selection algorithm rather than a kernel combination algorithm (hence may not suit multi-modal applications!). One of the reasons is the fact that we are essentially employing l_1 regularization over the kernel weights (i.e., λ's). Hence most of λ's will be zero at optimality. As soon as we note this, a little thought will motivate formulations which employ $l_p, p \geq 1$ regularization over the kernel weights [Cortes et al., 2009, Kloft et al., 2009]. These ofcourse will promote non-sparse combination of kernels and are shown to achieve varied degrees of success in applications. Nevertheless more principled ways of inducing sparsity and non-sparsity can be devised by exploring another view of the MKL problem:

A "primal" view of the problem was sought. We noted a formulation which is motivated from SVMs itself (rather than from trace-margin bound) which is equivalent to the MKL formulation. We will discuss this further in the next lecture.

Lecture 16

16.1 Summary

We began by discussing the primal view of the MKL problem, which is motivated purely from an alternate regularization (l_1 norm of l_2 norms) in context of SVM. Using a lemma (which we called as "lambda trick") we presented a dual of this formulation which turned out to be the (l_1)MKL. It was interesting to note that a purely optimization result naturally lead to MKL which was motivated from the trace-margin bound!

Once this interesting equivalence was in place, it was easy to see a more generic equivalence: primal formulation with an $l_q \equiv l_{2p/(p+1)}$ norm based regularizer is equivalent to l_p-MKL ($p \geq 1$). Now clearly, $q \in [1, 2]$ for $p \geq 1$. Hence we noted that one can obtain more generic MKL formulations by playing around with the primal regularizers (than with the dual's in some sense). Once this is established, researchers started to play around with norms leading to i) MKL formulations capable for handling multi-modal tasks [Nath et al., 2009, Aflalo et al., 2010] ii) MKL formulations leading to structured sparsity [Szafranski et al., 2008, Bach, 2008]. We concluded this lecture with a discussion on Nath et al. [2009], Aflalo et al. [2010] and reserved Bach [2008] for the subsequent lecture.

MKL for Multi-Modal tasks: The key idea was to choose a mixed-norm based regularizer in the primal MKL formulation which would support the prior information available in the case of multi-modal tasks. A particular dual of this formulation was noted, which again had nice interpretation of an SVM with weighted kernel — now weights exist for each mode of description of data as well as for the kernels. It was also noted that provably convergent and highly scalable algorithms based on Mirror-descent exist for solving this formulation (see Aflalo et al. [2010] for details). Some empirical results showing the benefits both in terms of generalization and scalability of the proposed methodology were discussed. The conclusion was that indeed good generalization can be achieved by clever regularizers which encode prior information regarding the task at hand. In the next lecture we will see another clever use of this idea for entirely a different purpose — of exploring large feature spaces using MKL [Bach, 2008].

16.2 Further Reading

- Details of the primal view and its equivalence to the MKL are detailed in Rakotomamonjy et al. [2008] (see also Bach et al. [2004], Rakotomamonjy et al. [2007]).

Lecture 17

17.1 Summary

The main goal of the lecture was to summarize the results in Bach [2008]. The goal of the work is to be able to perform l_1-MKL using exponentially large (in terms of no. input features) number of kernels efficiently. The key idea was to employ kernels which can be expressed as product of sums of kernels and can also be decomposed into a large sum of kernels. E.g., a specific polynomial kernel, ANOVA kernel etc. The next basic idea was to embed the large no. kernels in a DAG. The arrows in the DAG indicate: if the kernel at a node is not selected none of its descendents are not selected. In case of the polynomial and ANOVA kernels this turns out to represent that if a feature is not selected then none of the features constructed using this feature will be selected. Once this is in place few key theorems essentially lead to a simple iterative algorithm which solves a usual MKL problem (with few no. kernels) using say Mirror-descent or `simpleMKL` at each step. The computational effort is shown to be polynomial in the no. of selected kernels for such a setting.

Lecture 18

18.1 Summary

This lecture introduces the paradigm of Multi-Task Learning (MTL) [Caruana, 1997]. In contrast to a regular learning problem, in case of MTL, the goal is to simulataneously learn concepts regarding multiple *related* tasks. It was noted that different ways of specificfying task relatedness leads to different learning scenarios (including multi-modal, multi-class learning applications).

Following Evgeniou et al. [2005] it was noted that one way to encode task relatedness is to change the regularizer from $\mathbf{w}^\top\mathbf{w}$ to say, $\mathbf{w}^\top\mathbf{Q}\mathbf{w}$ for some psd $\mathbf{Q}$. E.g., of $\mathbf{Q}$ was represented and it was noted that $\mathbf{Q} = \mathbf{I}$ gives the scenario of unrelated tasks, in which case the whole excerise boils down to learning the tasks individually. It was also noted that playing with $\mathbf{Q}$ is equivalent to playing with the kernel. For related tasks, the kernel for examples from different tasks would have non-zero entries; whereas for unrelated tasks they would be zero.

We then looked at another way of playing with the regularizer which introduces yet another way of specifying task relatedness — all the tasks share the the same low-dimensional feature space (see eqn.(5) in Argyriou et al. [2008a]; this paper will be discussed in detail in the next lecture). This formulation was shown to be a special case of the MKL formulation; and hence can be efficiently solved using mirror-descent (see supplementary for details). For sake of completeness we noted few related learning theoretic bounds (refer Maurer [2006] for details).

Lecture 19

19.1 Summary

The paper by Argyriou et al. [2008a] was the main topic of discussion in this lecture. The key idea was to learn a low-dimensional subspace of features which a important for all the tasks. The notion of task relatedness was ofcourse again that all tasks share same feature-representation; however in contrast to the formulation discussed during end of last lecture, here the features are also learnt. Using the lambda trick and the formulation was written in a dual form whose interpretation was simple: learn a kernel which is $k(\mathbf{x}, \mathbf{y}) = \mathbf{x}^\top \mathbf{Q}\mathbf{y}, \mathbf{Q} \succeq 0$ using data from all the tasks. We noted that mirror descent algorithm can be applied since the feasibility set was essentially a spectrahedron (which is a well-known case). However Argyriou et al. [2008a] choose to re-write the formulation in the primal form itself in such a way that applying an alternating minimization algorithm was the obvious thing to do. For a fixed $\mathbf{Q}$, the problem is solving SVMs for each task individually and for fixed feature-weightings, the problem in $\mathbf{Q}$ interestingly had a close form solution in terms of EVD of the matrix of feature-weightings. So per-iteration computational cost is essentially $T * O_{SVM} + O(d^3)$.

Since the problem in $\mathbf{Q}$ has a closed form solution we can actually re-write the formulation eliminating $\mathbf{Q}$ — leading to a trace-norm regularization problem. We had a brief discussion of matrix norms and noted the analogy between l_1 norm (promotes sparsity in entries of vector) and trace-norm (promotes sparsity in singular values i.e., low rank matrix rather than in entries)! We also noted that trace-norm is "best" convex relaxation of the rank constraint (which is like l_0-norm). Interestingly the trace-norm minimization problem can be solved very efficiently [Ji and Ye, 2009]. At each iteration it requires to compute an SVD of a $d \times T$ matrix and hence per-step computational cost is $O(min(d, T)^3 + T^2 d + d^2 T)$. After briefing discussing the main idea behind this optimization methodology (which is originally due to Nesterov [2005]), we concluded the lecture with some interesting simulation results.

19.2 Further Reading

- Some linear algebra stuff can be refreshed using Saketh [2009].
- Some references for the Nesterov's method: Nesterov [2003], Beck and Teboulle [2009]
- Details of mirror-descent with spectrahedron set-up are in Nemirovski [2005] (sec. 5.4.1, 5.4.2, 5.5.1)

Lecture 20

20.1 Summary

In this lecture we completed a left over proof in Argyriou et al. [2008a] regarding the case of non-invertible $\mathbf{Q}$. The proof essentially follows from knowledge of EVD. We then had a look at kernelizing the formulation under discussion. The key idea is to come-up with an extension of the representer theorem for this case. It was clear that if the formulation involved a Frobenius norm (instead of a trace-norm) then the representer theorem is immediate — showing that the feature loadings of each task ($\mathbf{w}_t$) are linear combinations of training datapoints of that (t^{th}) task. However this does not hold in our case because of the presence of the trace-norm (inplace of Frobenius norm). However using some results regarding matrix monotone functions [Bhatia, 1997] we were able to show that $\mathbf{w}_t$ are linear combinations of **all** training datapoints (across all tasks) — thus arriving at the representer theorem needed. Once this is clear, borrowing ideas from kernel PCA [Schölkopf et al., 1998] we noted a way to kernelize the formulation (refer Argyriou et al. [2008a] for all details). We also discussed related simulation results.

Once this formulation is in place (recall similarity of this formulation to l_1-MKL through the spectrahedron constraints), it is natural to think about extending the formulation to other shatten-norms using ideas similar to l_p-MKL. This is discussed in detail in Argyriou et al. [2007, 2009]. There are also works on corresponding representer theorems [Argyriou et al., 2008b]. This ends our discussion on Multi-task learning. We will begin with Robust Learning in the next lecture.

Lecture 21

21.1 Summary

We started by motivating the need for learning algorithms being robust to various kinds of noise/uncertainties present in the training datapoints (We postponed discussion on issues of handling uncertainty in test datapoints, labels and kernelization of robust algorithms). We argued that if no explicit information regarding the underlying uncertainties in the datapoints is provided, then SVMs are good enough in the sense that they are already robust to noise. In particular, we showed that the (hard-margin) SVM classifier obtained with nominal training datapoints and that obtained using training datapoints with spherical noise balls around them are the same! Infact, this is happening is because 2-norm is the dual-norm of itself i.e., $\max_{\|\mathbf{x}\|_2 \leq 1} \mathbf{w}^\top \mathbf{x} = \|\mathbf{w}\|_2$. From a regularization point of view, l_2-norm based regularization for linear models hence provides robust solutions i.e., small perturbations of the data do not effect the model. As a passing note we formally defined dual norms and some relevant formulae. We also described the principle of robust optimization and noted that the above (and subsequent) discussions follow the ideas of robust optimization. According to this principle, the constraints of the optimization problem are ensured to be satisfied for all possible values of the parameters (here datapoints) — thereby the variables (hyperplane parameters in our case) remain feasible even for adverse values of the parameters (datapoints in our case). Thus in some sense, we are optimizing (designing) for the worst-case scenario.

With this background we considered scenarios where explicit (partial) information regarding the uncertainties is known. We considered the following cases: i) sperical noise (with given radii of noise balls) ii) elliptical noise iii) Extreme values of features are known i.e., rectangular noise iv) mean and covariance of noise distribution is known and noise distribution is Normal v) mean and covariance of noise distribution is alone known vi) extreme values, mean, variance of feature values are known. Note that these are listed in increasing order of amount of partial information available.

Cases i), ii), iii) easily follow from above discussion. i) and ii) can be posed as SOCPs [Bi and Zhang, 2004] whereas iii) as QP [Ghaoui et al., 2003]. iv) is a classical result and is discussed in many books (e.g., sec. 2.6 in [Lobo et al.]) and follows from posing the SVM problem as a chance-constrained program (CCP [Ben-Tal et al., 2009]). CCPs are in general hard problems to solve. However, interestingly, iv) is similar to ii) with only difference being interpretation of the radius of ellipsoid. In case of iv), radius has nice interpretation and is dependent on the probability η with which we wish to satisfy the feasibility (classification) constraints. If $\eta = 1$, radius turns out to be ∞, which is intuitive and when $\eta = 0.5$, the constraints are equivalent to restricting the means (alone) of datapoints to lie on the correct side of the hyperplane. For $\eta < 0.5$ (which is unintersting), the formulation is no more convex. In the next lecture we will discuss formulations for the remaining scenarios.

21.2 Further Reading

- Ben-Tal et al. [2009] is an excellent book on robust optimization describing state-of-the-art techniques; however is very technical and may be difficult to read.

- Lobo et al. is a good manuscript describing problems which can be posed as SOCPs.

Lecture 22

22.1 Summary

In this lecture we concentrate on scenario v). The key aspect in iv) which helped us re-write chance-constraints with usual constraints is the fact that the distribution of any linear combination of jointly Normal rvs, when normalized i.e., after mean substraction and sclaing by std.div., is the standard Normal distribution (which is independent of the optimization variables!). Moreover, the resultant constraints conviniently turn out to be convex. However this cannot be expected to happen with other distributions. In addition, for scenario v), the noise distribution is not even known. Such generic chance-constraints are indeed very difficult to handle and usually are highly non-convex.

One way out is to further the ideas of robust optimization and design/solve the optimization problem for some "worst-case" distribution i.e., come up with an (tight) upper bound on the required probability and contrain that the bound itself is greater than η. Such constraints are tighter than required (hence conservative) however, it is in some sense the best we can do if it is gauranteed that the bound is attained for some distribution. Ofcourse the bound must be chosen depending on the partial information known about the noise distribution.

In scenario v), mean and covariance of noise distribution (alone) is known and hence it is wise to employ a one-sided version of Chebyshev's inequality[1] (which is tight i.e., there exits a distribution with the given moments for which the inequality is active). We presented a short proof of this inequality and used it to derive a relevant bound in our case. Interestingly, the resultant constraint (bound greater than η) is convex and moreover similar to case iv)! Hence the problem is again of maximum-margin classification of ellipsoids! Only difference is in the radius term which in this case turns out to be $\kappa = \sqrt{\frac{\eta}{1-\eta}}$. Again, note

[1] See http://www.btinternet.com/~se16/hgb/cheb.htm

that, when $\eta = 1$, radius is ∞ and when $\eta = 0$, radius is 0 and the problem is equivalent to maximum-margin classification of mean datapoints.

In case of vi), additionally extreme values of features are given. The easiest way of incorporating this additional information is by considering uncertainty region around each datapoint as intersection of ellipse (given by the Chebyshev's inequality) and the hyper-rectangle (formed using the extreme values of features). However one can do better if "Bernstein" inequalities [Ben-Tal et al., 2009] are employed.

Now in case of most of the above discussed scenarios, the moments need to be estimated (for example, in case of micro-array datasets, replicates of each datapoint which represent different runs of the same experiment are provided and the moments can thus be estimated from them). However now the formulations need to made robust to moment estimation errors! Statisticians have put some effort in determining how far an empirical estimate of moments can be from the true ones. Atleast in case of Normal distributions, these confidence regions can be analytically computed. Now given these confidence regions, in which the true moments can lie around the estimated ones, following the principle of robust optimization, one can insist on the constraints being satisfied for any pair of moments in these confidence regions. Thus most of discussed formulations can also be made robust moment estimation errors.

Finally, we concluded with a brief discussion of some relevant results.

22.2 Further Reading

- Formulations for scenario v) are discussed in Bhattacharyya et al. [2004], Shivaswamy et al. [2006]
- Formulations for scenario vi) are discussed in Bhadra et al. [2009], Ben-Tal et al.
- Some reading on confidence regions for true moments: Arnold and Shavelle [1998]

Bibliography

J. Aflalo, A. Ben-Tal, C. Bhattacharyya, J. Saketha Nath, and S. Raman. Variable Sparsity Kernel Learning — Algorithms and Applications. In submission. Available online at `http://mllab.csa.iisc.ernet.in/vskl.html`, 2010.

M. A. Aizerman, E. M. Braverman, and L. I. Rozonoer. Theoretical Foundations of the Potential Function Method in Pattern Recognition Learning. *Automation and Remote Control*, 25:821–837, 1964.

A. Argyriou, C. A. Micchelli, M. Pontil, and Y. Ying. A Spectral Regularization Framework for Multi-Task Structure Learning. In *Proceedings of NIPS*, 2007.

A. Argyriou, T. Evgeniou, and M. Pontil. Convex Multi-Task Feature Learning. *Machine Learning*, 73(3):243–272, 2008a.

A. Argyriou, C. A. Micchelli, and M. Pontil. When Is There a Representer Theorem ? Vector versus Matrix Regularizers. *Journal of Machine Learning Research*, 2008b.

A. Argyriou, C. A. Micchelli, and M. Pontil. On Spectral Learning. *Journal of Machine Learning Research*, 2009.

Arnold and Shavelle. Joint Confidence Sets for the Mean and Variance of a Normal Distribution. *The American Statistician*, 52(2):133–140, 1998.

F. Bach. Exploring Large Feature Spaces with Hierarchical Multiple Kernel Learning. In *Advances in Neural Information Processing Systems (NIPS)*, 2008.

F.R. Bach, G.R.G. Lanckriet, and M.I. Jordan. Multiple Kernel Learning, Conic Duality, and the SMO Algorithm. In *Proceedings of the 21st International Conference on Machine Learning*, 2004.

P. L. Bartlett and S. Mendelson. Rademacher and Gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3: 463–482, 2002.

P. L. Bartlett and J. Shawe-Taylor. Generalization Performance of Support Vector Machines and Other Pattern Classifiers. *Advances in Kernel Methods — Support Vector Learning*, pages 43–54, 1999.

A. Beck and M. Teboulle. A Fast Iterative Shrinkage-thresholding Algorithm for Linear Inverse Problems. *SIAM Journal on Imaging Sciences*, 2:183–202, 2009.

Amir Beck and Marc Teboulle. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Operations Research Letters*, 31:167–175, 2003.

Shai Ben-David. ECE 695 "Statistical Learning Theory" Lecture Notes — Lecture 4. Available at `http://www.csl.cornell.edu/courses/ece695n/ece695_24sep2003.pdf`, 2003.

A. Ben-Tal, S. Bhadra, C. Bhattacharyya, and J. Saketha Nath. Chance constrained uncertain classification via robust optimization. *Mathematical Programming Series B special issue on Machine Learning (Accepted paper).*

A. Ben-Tal, L. El. Ghaoui, and A. Nemirovski. *Robust Optimization*. Princeton University Press, 2009.

Aharon Ben-Tal, Tamar Margalit, and Arkadi Nemirovski. The Ordered Subsets Mirror Descent Optimization Method with Applications to Tomography. *SIAM Journal of Optimization*, 12(1):79–108, 2001.

Kristin P. Bennett and Eric. J. Bredensteiner. Duality and Geometry in SVM Classifiers. In *Proceedings of the International Conference on Machine Learning*, pages 57–64, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc. ISBN 1-55860-707-2.

D. P. Bertsekas. *Nonlinear Programming*. Athena Scientific, 2 edition, 1999.

S. Bhadra, J. Saketha Nath, A. Ben-Tal, and C. Bhattacharyya. Interval data classification under partial information: A chance-constraint approach. In *Proceedings of the PAKDD conference*, 2009.

R. Bhatia. *Matrix Analysis*. Springer-Verlag, 1997.

Chiranjib Bhattacharyya, P. K. Shivaswamy, and Alex J. Smola. A second order cone programming formulation for classifying missing data. In *Advances in Neural Information Processing Sytems*, 2004.

Jinbo Bi and Tong Zhang. Support Vector Classification with Input Data uncertainty. In *Advances in Nueral Information Processing Systems*, 2004.

B. E. Boser, I. M. Guyon, and V. N. Vapnik. A Training Algorithm for Optimal Margin Classifiers. In *Proceedings of the Annual ACM Workshop on Computational Learning Theory*, pages 144–152, 1992.

Stephane Boucheron, Gabor Lugosi, and Olivier Bousquet. Concentration Inequalities. *Advance Lectures in Machine Learning*, pages 208–240, 2004.

Olivier Bousquet, Stephane Boucheron, and Gabor Lugosi. Introduction to Statistical Learning Theory. *Advanced Lectures on Machine Learning Lecture Notes in Artificial Intelligence*, 3176:169–207, 2004.

Stephen Boyd and Lieven Vandenberghe. *Convex Optimization.* Cambridge University Press, 2004.

C. J .C. Burges. A Tutorial on Support Vector Machines for Pattern Recognition. *Data Mining and Knowledge Discovery*, 2(2), 1998.

C. J. C. Burges and David J. Crisp. Uniqueness of the svm solution. In *Advances in Neural Information Processing Systems*, 2000.

C. J. C. Burges and B. Schölkopf. Improving the accuracy and speed of support vector learning machines. In *Proceedings of the 9th NIPS Conference*, pages 375–381, 1997.

Rich Caruana. Multitask learning: A knowledge-based source of inductive bias. *Machine Learning*, 28:41–75, 1997.

O. Chapelle and A. Rakotomamonjy. Second Order Optimization of Kernel Parameters. In *NIPS Workshop on Automatic Selection of Optimal Kernels*, 2008.

O. Chapelle, V. N. Vapnik, O. Bousquet, and S. Mukherjee. Choosing Multiple Parameters for Support Vector Machines. 46(1–3):131–159, 2002.

C. Cortes and V .N. Vapnik. Support Vector Networks. 20:273–297, 1995.

C. Cortes, M. Mohri, and A. Rostamizadeh. L2 regularization for learning kernels. In *Proceedings of Uncertainty in Artificial Intelligence*, 2009.

Harris Drucker, Chris J.C. Burges, Linda Kaufman, Alex Smola, and Vladimir Vapnik. Support vector regression machines. In *Advances in Neural Information Processing Systems*, volume 9, pages 155–161. MIT Press, 1996.

T. Evgeniou, C. A. Micchelli, and M. Pontil. Learning Multiple Tasks with Kernel Methods. *Journal of Machine Learning Research*, 6:615–637, 2005.

L. El. Ghaoui, G. Lanckriet, and G. Natsoulis. Robust Classification with Interval Data. Technical report, EECCS Department, University of California, Berkeley, 2003.

A. E. Hoerl. Application of ridge analysis to regression problems. *Chemical Engineering Progress*, 58:54–59, 1962.

V. V. Ivanov. *The Theory of Approximate Methods and Their Application to the Numerical Solution of Singular Integral Equations*. Nordhoff International, 1976.

Tony Jebara, Risi Kondor, and Andrew Howard. Probability product kernels. *Journal of Machine Learning Research*, 5:819–844, 2004. ISSN 1532-4435.

Shuiwang Ji and Jieping Ye. An Accelerated Gradient Method for Trace Norm Minimization. In *Proceedings of the International Conference on Machine Learning*, 2009.

S. S. Keerthi, S. K. Shevade, C. Bhattacharyya, and K. R. K. Murthy. A fast iterative nearest point algorithm for support vector machine classifier design. *IEEE Transactions on Neural Networks*, 11:124–136, 2000.

S. S. Keerthi, S. K. Shevade, C. Bhattacharyya, and K. R. K. Murthy. Improvements to platt's smo algorithm for svm classifier design. *Neural Computation*, 13(3):637–649, 2001. ISSN 0899-7667. doi: http://dx.doi.org/10.1162/089976601300014493.

S. S. Keerthi, O. Chapelle, and D. DeCoste. *Journal of Machine Learning Research*, 7:1493–1515, 2006.

M. Kloft, U. Brefeld, S. Sonnenburg, P. Laskow, K-R. Mueller, and A. Zien. Efficient and Accurate Lp-Norm MKL. In *Advances in Neural Information Processing Systems*, pages 997–1005, 2009.

V. Kolchinskii, D. Panchenko, and F. Lozano. Further Explanation of the Effectiveness of Voting Methods: The game between margins and weights. In *Proceedings of COLT*, 2001.

G.R.G. Lanckriet, N. Cristianini, P. Bartlett, L. El Ghaoui, and M.I. Jordan. Learning the Kernel Matrix with Semidefinite Programming. In *Proceedings of the 19th International Conference on Machine Learning*, 2002.

G.R.G. Lanckriet, N. Cristianini, P. Bartlett, L. El Ghaoui, and M.I. Jordan. Learning the Kernel Matrix with Semidefinite Programming. *Journal of Machine Learning Research*, 5:27–72, 2004.

M. Lobo, L. Vandenberghe, S. Boyd, and H. Lebret.

David G. Luenberger and Yinyu Ye. *Linear and Nonlinear Programming*. Springer, 3 edition, 2008.

Andreas Maurer. Bounds for Linear Multi-Task Learning. *Journal of Machine Learning Research*, 7:2006, 2006.

C. McDiarmid. On the methods of bounded differences. *Surveys in Combinatorics*, pages 148–188, 1989.

Shahar Mendelson. A Few Notes on Statistical Learning Theory. *Advanced Lectures on Machine Learning*, pages 1–40, 2003.

J. Mercer. Functions of positive and negative type and their connection with the theory of integral equations. *Philosophical Transactions of the Royal Society A*, 209:415–446, 1909.

C. A. Micchelli. Algebraic Aspects of Interpolation. In *Proceedings of Symposia in Applied Mathematics*, volume 36, pages 81–102, 1986.

V. A. Morozov. *Methods for Solving Incorrectly Posed Problems*. Springer-Verlag, 1984.

J. Saketha Nath, G Dinesh, S Raman, Chiranjib Bhattacharyya, Aharon Ben-Tal, and Ramakrishnan K.R. On the algorithmics and applications of a mixed-norm based kernel learning formulation. In *Advances in Neural Information Processing Systems 22*, pages 844–852, 2009.

A. Nemirovski. Lectures on Modern Convex Optimization. `http://www.isye.gatech.edu/faculty-staff/profile.php?entry=an63`, 2005.

Y. Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*. Kluwer Academic Publishers, 2003.

Y. Nesterov. Smooth minimization of non-smooth functions. *Mathematical Programming*, 103:127–152, 2005.

John. C. Platt. Fast training of support vector machines using sequential minimal optimization. pages 185–208, 1999.

A. Rakotomamonjy, F. Bach, S. Canu, and Y Grandvalet. SimpleMKL. *Journal of Machine Learning Research*, 9:2491–2521, 2008.

Alain Rakotomamonjy, Francis Bach, Stéphane Canu, and Yves Grandvalet. More efficiency in multiple kernel learning. In *ICML '07: Proceedings of the 24th international conference on Machine learning*, pages 775–782, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-793-3. doi: http://doi.acm.org/10.1145/1273496.1273594.

Saketh. Lecture Notes for CS723. Available at `http://www.cse.iitb.ac.in/saketh/teaching/cs723.html`, 2009.

B. Schölkopf, A. Smola, and K.-R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10(5):1299–1319, 1998.

Bernhard Schölkopf and Alex Smola. *Learning with Kernels.* MIT press, Cambridge, 2002.

Bernhard Schölkopf, A. J. Smola, R. C. Williamson, and P. L. Bartlett. New support vector algorithms. *Neural Computation*, 12:1207–1245, 2000.

Bernhard Schölkopf, Ralf Herbrich, and Alex J. Smola. A generalized representer theorem. In *COLT '01/EuroCOLT '01: Proceedings of the 14th Annual Conference on Computational Learning Theory and and 5th European Conference on Computational Learning Theory*, pages 416–426, London, UK, 2001. Springer-Verlag. ISBN 3-540-42343-5.

J. Shawe-Taylor and N. Cristianini. Data-dependent Structural Risk Minimization. In *Advances in Neural Information Processing Systems*, 1998.

J. Shawe-Taylor, P. L. Bartlett, R. C. Williamson, and M. Anthony. Structural Risk Minimization Over Data-Dependent Hierarchies. *IEEE Transactions on Information Theory*, 44(5):1926–1940, 1998.

John Shawe-Taylor and Nello Cristianini. *Kernel Methods for Pattern Analysis.* Cambridge University Press, 2004.

P .K. Shivaswamy, Chiranjib Bhattacharyya, and Alexander J. Smola. Second order cone programming approaches for handling missing and uncertain data. *Journal of Machine Learning Research*, 7:1283–1314, 2006.

M. Sion. On General Minimax Theorem. *Pacific Journal of Mathematics*, 1958.

Alex J. Smola and Bernhard Schölkopf. A tutorial on support vector regression. *Statistics and Computing*, 14(3):199–222, 2004.

Roman Smolensky. Well-Known Bound for the VC-Dimension Made Easy. *Computational Complexity*, 6:299–300, 1997.

Soren Sonnenburg, Gunnar Ratsch, Christin Schafer, and Bernhard Scholkopf. Large Scale Multiple Kernel Learning. *Journal of Machine Learning Research*, 7:1531–1565, 2006.

E. D. Sontag. VC dimension of neural networks. *Neural Networks and Machine Learning*, pages 69–95, 1998.

M. Szafranski, Y. Grandvalet, and A. Rakotomamonjy. Composite Kernel Learning. In *ICML '08: Proceedings of the 25th international conference on Machine learning*, pages 1040–1047, 2008.

A. N. Tikhonov and V. Y. Arsenin. *Solution of Ill-Posed Problems.* Winston, 1977.

L. Vandenberghe and S. Boyd. Semidefinite Programming. *SIAM Review*, 38(1): 49–95, 1996.

V. Vapnik and A. Chervonenkis. The necessary and sufficient conditions for consistency in the Empirical Risk Minimization method. *Pattern Recognition and Image Analysis*, 1(3):283–305, 1991.

V. N. Vapnik. *Statistical Learning Theory.* John Wiley and Sons, Inc., 1998.

V. N. Vapnik and A. Chervonenkis. On the Uniform Convergence of Relative Frequencies of Events to their Probabilities. *Theory of Probability and its Applications*, 16(2):264–280, 1971.

Zenglin Xu, Rong Jin, Irwin King, and Michael R. Lyu. An Extended Level Method for Multiple Kernel Learning. In *Advances in Neural Information Processing Systems*, 2008.